Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 01-043

Improve Precategorized Collection Retrieval by Using Supervised Term Weighting Schemes

Ying Zhao and George Karypis

December 10, 2001

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
21 OCT 2001

2. REPORT TYPE

3. DATES COVERED
-

4. TITLE AND SUBTITLE
Improve Precategorized Collection Retrieval by Using Supervised Term Weighting Schemes

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Army Research Office,PO Box 12211,Research Triangle Park,NC,27709-2211

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**10**	

Improve Precategorized Collection Retrieval by using supervised term weighting schemes [*]

Ying Zhao and George Karypis

University of Minnesota, Department of Computer Science/Army HPC Research Center
Minneapolis, MN 55455
Technical Report #01-43

{yzhao,karypis}@cs.umn.edu

October 21, 2001

Abstract

The emergence of the world-wide-web has led to an increased interest in methods for searching for information. A key characteristic of many of the online document collections is that the documents have predefined category information, for example, the variety of scientific articles accessible via digital libraries (e.g., ACM, IEEE, etc.), medical articles, news-wires, and various directories (e.g., Yahoo, OpenDirectory Project, etc.). However, most previous information retrieval systems have not taken the pre-existing category information into account. In this paper, we proposed weight adjustment schemes based upon the category information in the vector-space model, which are able to select the most content specific and discriminating features. Our experimental results on TREC data sets show that the pre-existing category information does provide additional beneficial information to improve retrieval. The proposed weight adjustment schemes perform better than the vector-space model with the inverse document frequency (IDF) weighting scheme when queries are less specific. The proposed weighting schemes can also benefit retrieval when clusters are used as an approximation to categories.

[*] This work was supported by NSF CCR-9972519, EIA-9986042, ACI-9982274, by Army Research Office contract DA/DAAG55-98-1-0441, by the DOE ASCI program, and by Army High Performance Computing Research Center contract number DAAH04-95-C-0008.

1 Introduction

The emergence of the world-wide-web has led to an increased interest in methods for searching for information. An important characteristic of the online document collections is that more and more predefined category information is available, for example, digital libraries categories (e.g., ACM, IEEE, etc.), medical articles (e.g., Medline etc.) and web directories (e.g., Yahoo, OpenDirectory Project, Google etc.), In the meantime, text classification and organization has been extensively studied in both information retrieval and text mining literatures. However, there is little work done on combining the category information with traditional IR techniques to improve retrieval. Srinivasan [13, 14] combined category labels and original query terms to expand queries. The category labels (MeSH terms) indeed improve the quality of the retrieved information. However, the proposed method has requirements on class labels, which makes this approach hard to be generalized. Another set of techniques utilizing category information is supervised dimensionality reduction, which refers to the set of techniques that take advantage of class-membership information while computing the lower dimensional space. Examples of such techniques include a variety of feature selection schemes [1, 7, 9, 8, 17, 4, 16, 11, 10] that reduce the dimensionality by selecting a subset of the original features, techniques that create new features by clustering the terms [2], techniques based on local latent semantic indexing [5, 12], and techniques based on supervised concept indexing [6].

All these previous approaches only used a limited portion of the additional information among documents that the pre-existing category information provides us. Presumably, the manually constructed category information provides additional thematic information that can benefit further term studies and enhance retrieval.

In this paper, we explored an alternative way to utilize category information by adjusting term weights based upon the term's distribution among categories. We presented a normalized entropy (NE) method to determine the category specificity of each term, from which we derived two supervised term weighting schemes. The evaluation results on TREC data sets show that the proposed schemes outperform the traditional *IDF* scheme significantly when the queries contain more than a few specific terms and achieve competitive results on short and well-defined queries. The experimental results also show that the proposed term weighting schemes can still benefit retrieval even when categories are approximated by clusters which are generated automatically.

2 Supervised Term Weighting Schemes

Our research of utilizing category information is motivated by analyzing the distribution of relevant documents across categories on various data sets. This analysis indicates that relevant documents tend to concentrate into a few categories. Table 1 shows such analysis on two data sets: FT and LATimes, which compares the observed distribution of relevant documents and the expected distribution if they were distributed uniformly. We calculated the entropy of the relevant documents distribution among categories as the following:

$$entropy(t_i) = -\sum_{c \in C} P(c|t_i) \log P(c|t_i)$$

where C is the set of categories. The expected entropy values were calculated based upon the underlying category distribution assuming the relevant documents were distributed uniformly across the collection. In Table 1, we present the observed mean and standard deviation of the entropy, the expected value and t-value of the difference for both FT data set and LATimes data set. The significant levels are all above 0.01, which indicates that the distribution of relevant documents across categories is far from uniform.

Table 1: Comparison of observed relevant documents distribution against expected if relevant documents were distributed uniformly

Data set	# of queries	mean	s.d.	expected	t-value
LATimes(>20)	54	1.64	.40	2.39	13.5
FT(>50)	33	1.83	.55	3.869	20.88

This observed characteristic of relevant documents indicates that relevant documents may contain category specific terms, which make those relevant documents belong to particular categories. If we can construct a method that is able to distinguish category specific terms from other terms, then somehow we associate thematic meanings with terms, which allows us to be able to emphasize the terms that represent the content of documents and categories.

Table 2: An example: Query 360

$< num >$ Number: 360
$< title >$drug legalization benefits
$< desc >$ Description:
What are the benefits, if any, of drug legalization

(a)

	drug	benefit	legal
IDF	2.7222	2.9139	2.8008
NE	0.1326	0.1374	0.1979
# of relv docs	48	6	44

(b)

the IDF Scheme

Doc ID	Relv?	Drug	Benefit	Legal
29493	r	1.0000	0.6250	0.6250
27426	r	1.0000	0.5238	0.7619
115777	r	1.0000	0.5100	0.7200
24656	r	1.0000	0.5500	0.6000
54526	n	0.5333	1.0000	0.5667
91955	n	0.7778	0.6111	0.7222
5509	n	1.0000	0.5357	0.5714

(c)

the NE Scheme

Doc ID	Relv?	Drug	Benefit	Legal
27426	r	1.0000	0.5238	0.7619
115777	r	1.0000	0.5100	0.7200
29493	r	1.0000	0.6250	0.6250
117947	r	1.0000	0	1.0000
29495	r	1.0000	0	1.0000
91955	n	0.7778	0.6111	0.7222
24656	r	1.0000	0.5500	0.6000

(d)

To find out the terms representing content according to category information, we developed a measure of term specificity based upon a term's distribution among categories. If a term only appears in one or few categories, it has high certainty with respect to these categories. On the other hand, if a term appears across most of the categories, then the possibility that the term represents the content of any category is low. We used normalized entropy as the measure of this category information uncertainty, which is defined as follows:

$$NE(t_j) = -\sum_{i=1}^{M} p_{ij} \log p_{ij}, \tag{1}$$

where M is the total number of categories, and p_{ij} is given by $\frac{P(t_j|c_i)}{\sum_{k=1}^{M} P(t_j|c_k)}$, with $P(t_j|c_i)$ equals the number of documents containing the term t_j in the category c_i divided by the total number of documents in the category c_i.

The normalized entropy (NE) defined above eliminates the effect of the variation of category sizes. The portion of the documents containing the term t_j in the category c_i is more suitable to represent the distribution of the term then the absolute number of documents containing the term t_j. When all the categories have similar sizes, $p_{ij} = P(c_i|t_j)$.

In the rest of this section, we will present two term weighting schemes: the normalized entropy (NE) scheme and the combined NE and *IDF* scheme, which derive the term weights based upon the normalized entropy described above. We refer them as supervised term weighting schemes.

2.1 The Normalized Entropy Scheme

In the normalized entropy (NE) scheme, the weight of the term t_j is given by

$$w_{t_j} = NE_{max} - NE(t_j),$$

where NE_{max} is the maximum normalized entropy of all the terms and NE_{t_j} is defined in Equation 1. The normalized entropy (NE) scheme will give high weights to the terms that are specific to a few categories.

We present an example to illustrate how the normalized entropy (NE) scheme is able to emphasize content specific terms, which the *IDF* scheme fails to identify. The example is to perform the query number 360 on the LATimes data set.

Table 2 (a) describes the content of the query. Table 2 (b) shows the different weights assigned by the *IDF* scheme and the *NE* scheme to the three terms in the query 360 and the number of relevant documents that really contain that term. The *IDF* scheme gives similar weights to all the three terms, which means they occur in the collection with similar frequency. However, they do behave differently according to the *NE* scheme. "legalization" is more category specific then the others, which represents an important component of the relevant documents. Instead of giving the highest weight to "benefit" as the *IDF* scheme, the *NE* scheme is able to emphasize the content specific term “legalization”.

Table 2 (c) and Table 2 (d) list the first seven documents retrieved by the *IDF* scheme and the *NE* scheme, respectively. Each entry contains the retrieved document ID, whether the document is relevant or not, followed by the normalized term frequency for each query term, where the normalized term frequency will be defined in details in Section 3. By giving more weight to the term “legalization” and less weight to the term “benefit”, the *NE* scheme is able to retrieve relevant documents that do not contain the term “benefit” explicitly. This example illustrates one of the limitations of the *IDF* scheme: when all the terms occur in the collation with similar moderate frequency, the *IDF* scheme can not further tell the difference based upon the term’s distribution.

2.2 The Combined NE and IDF Scheme

The normalized entropy (NE) scheme has nice properties to emphasize the content specific terms. However, it also has limitations: the content specific terms can lead us to the best matching categories, but if those terms are the common terms in the categories, then those terms actually have limited discriminating power. They are not able to further distinguish relevant documents from the irrelevant documents in the same categories. We can understand this more clearly by comparing NE with IDF in the following table.

Table 3: Entropy vs. IDF

	high Entropy	low Entropy
high IDF	Good	Good/Bad
low IDF	Good/Bad	Bad

Table 3 shows the four possible combinations of NE and IDF values for a term. The terms with high NE and low IDF are helpful for locating categories, but are not able to future eliminate the irrelevant documents in the same categories. By combining IDF and NE, we can avoid to emphasize too much on those terms.

In addition, the terms with low NE and high IDF will be good terms to retrieve relevance if the information need covers larger number of categories, in which case, the use of NE as term selecting criteria will de-emphasize those good terms. On the other hand, those terms will be bad if the information need only covers small number of categories, in which case, the use of NE will correct the wrong emphasis caused by the *IDF* scheme. This nature of NE leads to a variation of the performance, i.e., the improvement gained by utilizing category information tends to increase when the category structure matches the information need better.

We developed the combined scheme to further improve the normalized entropy (NE) scheme by combining IDF and NE as follows:

$$w_{t_j} = ((NE_{max} - NE(t_j)) * IDF_j)^{\alpha},$$

where NE_{max} is the maximum normalized entropy of all the terms, IDF_j is the IDF value of the term t_j and α is the scaling power.

The combined scheme will give high weights to the terms that are both category specific and infrequent. It is nature to have $\alpha = 1$, in which case, the combined scheme gives much higher weights to those category specific and infrequent terms than the other terms, which may result in a loss of information when queries only contain a few terms. To make the combined scheme have the same scale after transfermation, we choose another scaling power to be 0.5, since IDF and NE are both logarithmic functions. In the following evaluation section, we performed two trails of experiments with $\alpha = .5$ and $\alpha = 1$.

3 Experimental Evaluation

We selected three subcollections in the TREC collection that have category information as our experimal collections. The statistics of the three collections: Financial Times Limited (FT), Los Angeles Times (LATimes) and San Jose Mercury News (SJM) are shown in table 4. We derived category information from the IN field, SECTION field and DESCRIPT field, respectively. For each collection, we collected queries from TREC ad hoc topics that have relevent documents in that collection. Each query contains three parts: a title, a description and a narrative. We formed three types of query sets: long (t+d+n), medium (t+d) and short (t) by including all three parts, title and description parts, and title part only, respectively. The queries of SJM are from TREC-4 ad hoc topics, which only have the description part, thus only the medium type queries are presented. Table 4 shows the statistics of the query sets for each collection.

Table 4: Statistics for Data Sets and Queries

Data sets Statistics

Data set	# of docs	# of classes	min class size	max class size	avg class size
FT	210,158	83	26	14670	2632
LATimes	131,896	22	89	25837	5052
SJM	70,980	281	62	4456	492

Queries Statistics

Data set	# of queries	# avg relevant docs	avg length(t)	avg length(t+d)	avg length(t+d+n)
FT	141	34.63	2.41	7.9	21.21
LATimes	142	24.5	2.40	8.1	21.52
SJM	45	24.4	—	7.37	—

For each document in our data sets, we used a stop-list to remove common words and the words were stemmed using Porter's suffix-stripping algorithm. We represented each document i as a term vector using the popular vector-space model, in which the value for each term t_j was defined as follows:

$$w_{ij} = (0.5 + 0.5\frac{tf_{ij}}{max_j(tf_{ij})}) * w_{t_j},$$

where tf_{ij} is the term frequency of the term j in the document i, w_{t_j} is the term weight of the term t_j assigned by the IDF term weighting scheme or our supervised term weighting schemes. Each query also was represented as a term vector with the value for each term t_j as the normalized term frequency: $w_{ij} = 0.5 + 0.5\frac{tf_{ij}}{max_j(tf_{ij})}$, where tf_{ij} is the term frequency of the term j in the query i. Then we performed a dot-product similarity search between queries and every document in the data collection. We used uninterpolated average precision to measure the retrieval effectiveness among various term weighting schemes. Precision is defined as the ratio of the number of relevant documents retrieved over the number of retrieved documents. We compute the precision when each relevant document is retrieved, and the average of all precision values, the uninterpolated average precision, is used to measure the performance of the various term weighting schemes on a particular query.

3.1 Supervised Term Weighting Schemes

Our first set of experiments focused on comparing the performance of the various supervised term weighting schemes against that achieved by the traditional *IDF* scheme that does not take category information into account. These results are shown in Table 5 that shows the average precision achieved by the various term weighting schemes on the different queries for the three datasets. Each row in this table corresponds to a particular term weighting scheme and the various columns correspond to the different datasets and query types. In particular, the row labeled "IDF" corresponds to the traditional *IDF* scheme, the row labeled "NE" corresponds to the normalized entropy scheme, the row labeled "IDF*NE" corresponds to the combined scheme with $\alpha = 1.0$, and the scheme labeled "$\sqrt{\text{IDF*NE}}$" corresponds to the combined scheme with $\alpha = .5$. We use boldfaced fonts to highlight the best results for a particular query-dataset combination, and the entries that achieved the best overall results for each dataset are also underlined.

Table 5: Comparison of the average precision achieved by four term weighting schemes

	FT			LATimes			SJM
schemes	t+d+n	t+d	t	t+d+n	t+d	t	t+d
TF	0.092	0.151	0.228	0.071	0.148	0.224	0.135
IDF	0.205	0.267	**0.261**	0.163	0.227	**0.244**	0.176
NE	0.275	0.285	0.248	0.220	0.240	0.226	0.186
IDF*NE	**0.285**	0.277	0.240	**0.240**	0.229	0.221	0.176
$\sqrt{IDF * NE}$	0.269	**0.288**	0.255	0.210	**0.243**	0.240	**0.189**

Looking at the results in this table we can see that the performance of the various supervised term weighting schemes depends on the query type. In particular, for short queries (t), the proposed term weighting schemes tend to perform somewhat worse than the traditional *IDF* scheme. The traditional *IDF* scheme working well for short queries on TREC data sets is not surprising, since as explored by Greiff [3], for the query terms in the title and description parts of TREC queries, the document frequency of a term is a good approximation of the weight of evidence that a document contains that term is relevant. However, as the length of the queries increases all three proposed schemes lead to better results than those obtained by IDF. For the medium-size queries (t+d), they lead to improvements over IDF that range from 0% to 8%, whereas for long queries these improvements range from 28% to 48%. Among the three proposed schemes, we can see that the combined scheme with $\alpha = 1.0$ outperforms the rest for long queries, whereas the combined scheme with $\alpha = .5$ does the best on short and medium-length queries. Comparing the performance of the supervised term weighing schemes across the different datasets we can see that they lead to better results for FT and SJM and to somewhat worse results for the LATimes dataset. The best results for FT and SJM datasets across all types of queries and term-weighting schemes were achieved by the combined scheme with $\alpha = .5$, which also achieved the second best result for the LATimes dataset. The reason for the performance of the supervised term weighting schemes worse than that of the *IDF* scheme on the LATimes dataset may be due to the fact that the number of categories in the LATimes dataset was very small. Consequently, the supervised term weighting schemes can only provide limited additional discriminating power.

In general, the results in Table 5 suggest that the proposed supervised term weighting schemes are especially useful when the queries contain more than just a few terms. Such moderately large queries, quite often contain terms that are not very important in identifying relevant documents. As a result, by utilizing category information the proposed schemes can reduce the importance of these terms during the ranking calculations. The ability of the supervised term weighting schemes to de-emphasize such terms is also the reason why the combined scheme with $\alpha = 1.0$ does so well for large queries. Recall from Section 2.2 that when $\alpha = 1.0$, the combined scheme tends to assign high weights to terms that are both category specific and infrequent, and much smaller weight to the rest of the terms. The weight difference between these set of terms is much higher for this scheme than for any of the other supervised schemes. Now, in large queries, the number of non-critical terms will tend to be quite large, as a result the combined scheme with $\alpha = 1.0$ will end-up focusing on only a few of these terms and give very small weights to a large number of non-category specific and frequent terms, improving the overall retrieval results.

3.2 Unsupervised Term Weighting Schemes

Our second set of experiments focused on evaluating whether or not the proposed supervised term weighting schemes can also lead to retrieval improvements when the categories are automatically discovered using a clustering algorithm. The motivation behind this approach is that the documents within each cluster will most likely be part of the same topic, and as such the distribution of the terms in these clusters can be used to extract some additional thematic information, which can benefit retrieval. To this end, we used a vector-space bisecting K-means clustering algorithm [15] to cluster each one of the datasets into a certain number of clusters k, and then treat each of these clusters as a separate category and apply the various supervised term weighting schemes described in Section 2. We refer this set of term weighting schemes as unsupervised term weighting schemes.

Table 6 shows the average precision obtained for the different types of queries and datasets for different values of k. Similar to our earlier presentation, we boldfaced the entries that achieve the best results for each query-dataset combination and underlined the entries that achieved the best results for each dataset.

A number of interesting observations can be made by looking at the results of Table 6. First, as the number of clusters increases, the average precision achieved by the three term weighting schemes also tends to increase. For most types of queries and datasets, the highest values for each term weighting scheme are often achieved for 100–150 clusters. Second, the relative performance of the three supervised term weighting schemes for the different query types is quite similar to the relative performance achieved when the actual categories were used (Table 5). For short and medium queries, the combined scheme with $\alpha = 0.5$ tends to perform better than the other two, whereas for large queries the combined scheme with $\alpha = 1.0$ does the best. Third, for short and medium length queries, the relative performance of the combined scheme with $\alpha = 1.0$ approaches that of the combined scheme with $\alpha = 0.5$ as the number of clusters increases. This is due to the fact that as k increases, the normalized entropy of each term becomes small since the term is distributed across more clusters. Consequently, the weights of these terms using the NE approach become smaller and more uniform.

Finally, comparing the performance achieved by the clustering-based approaches to that obtained when the actual

Table 6: Comparison of the average precision achieved by unsupervised term weighting schemes with various number of clusters

FT Data Set

# of	t+d+n			t+d			t		
clusters	NE	IDF*NE	$\sqrt{IDF * NE}$	NE	IDF*NE	$\sqrt{IDF * NE}$	NE	IDF * NE	$\sqrt{IDF * NE}$
25	0.2219	0.2647	0.2167	0.2645	0.2718	0.2699	0.2463	0.2397	0.2550
50	0.2486	0.2805	0.2335	0.2747	0.2779	0.2780	0.2503	0.2411	**0.2589**
75	0.2517	**0.2844**	0.2353	0.2749	0.2783	0.2806	0.2507	0.2447	0.2571
100	0.2510	0.2815	**0.2354**	**0.2770**	**0.2800**	0.2799	0.2536	0.2471	0.2586
125	0.2503	0.2841	0.2347	0.2719	0.2796	0.2775	**0.2540**	0.2465	0.2579
150	**0.2548**	0.2832	0.2353	0.2767	0.2795	**0.2812**	0.2537	**0.2489**	0.2580

LATimes Data Set

# of	t+d+n			t+d			t		
clusters	NE	IDF*NE	$\sqrt{IDF * NE}$	NE	IDF*NE	$\sqrt{IDF * NE}$	NE	IDF * NE	$\sqrt{IDF * NE}$
25	0.1873	0.2410	0.1845	0.2278	0.2318	0.2329	0.2221	0.2172	0.2371
50	0.1996	0.2422	0.1921	0.2315	0.2330	0.2344	0.2277	0.2187	0.2387
75	0.2098	0.2498	0.1883	0.2364	0.2340	0.2356	0.2263	0.2192	0.2379
100	0.2227	0.2512	0.2016	0.2386	**0.2380**	0.2374	0.2301	0.2218	0.2422
125	0.2220	0.2548	0.1993	0.2375	0.2378	0.2408	**0.2314**	**0.2221**	**0.2427**
150	**0.2267**	**0.2578**	**0.2039**	**0.2449**	0.2353	**0.2422**	0.2312	0.2213	0.2426

SJM Data Set

# of clusters	NE	IDF*NE	$\sqrt{IDF * NE}$
25	0.1701	0.1657	0.1773
50	0.1772	**0.1763**	0.1802
75	0.1799	0.1724	0.1819
100	0.1811	0.1743	**0.1828**
125	0.1793	0.1728	0.1825
150	**0.1817**	0.1736	0.1824

category information was used we can see that there is little difference between the corresponding schemes. For all datasets and query types, the best clustering-based solution is usually within 0%–3% of that of the category-based solution. Moreover, the clustering-based solutions are considerably better (ranging from 0%–45%) than the results obtained by IDF for medium and long queries. This suggests that the proposed schemes can be used to improve the retrieval performance even in the absence of category information.

4 Conclusions and Future Work

In this paper, we explored an alternative way to utilize pre-existing category information: determining term weights based upon category specificity. We proposed two supervised term weighting schemes: the normalized entropy (NE) scheme and the combined scheme. The experimental results show that these two schemes substantially outperform the *IDF* scheme when queries are less specific and achieve similar results when queries are short and only contain specific terms. The results confirm that pre-existing category information indeed contains valuable thematic information to improve retrieval and the proposed schemes somehow are able to capture the hidden information. In addition, the proposed schemes can be extended to compute term weights based upon cluster specificity.

There are two issues need to be studied to further understand the proposed schemes. First, we would like to conduct similar data exploration analysis as stated in [3] to understand the relationship between the weight of evidence, IDF and NE when queries are less well-defined. This study would give us a more rigorous explanation why the proposed schemes work. Second, the scaling parameter of the combined scheme does change the behavior of the scheme. The scheme with scaling power of one works better with longer queries, whereas the scheme with scaling power of 0.5 works better with shorter queries. A full parameter study is needed to uncover the insight of this behavior and potentially can help to further improve the proposed schemes.

References

[1] H. Almuallim and T.G. Dietterich. Learning with many irrelevant features. In *Proc. of the Ninth International Conference on Machine Learning*, pages 547–552, 1991.

[2] L. Baker and A. McCallum. Distributional clustering of words for text classification. In *SIGIR-98*, 1998.

[3] W. R. Greiff. A theory of term weighting based on exploratory data analysis. *The Proceeding SIGIR'98*, 1998.

[4] S.J. Hong. Use of contextual information for feature ranking and discretization. *IEEE Transactions on Knowledge and Data Eng.*, 9(5):718–730, September/October 1997.

[5] D. Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *Proceedings of SIGIR*, pages 282–291, 1994.

[6] G. Karypis and E. H. Han. Fast supervised dimensionality reduction algorithm with applications to document categorization & retrieval. In *Proceedings of CIKM*, 2000.

[7] K. Kira and L.A. Rendell. A practical approach to feature selection. In *Proc. of the 10th International Conference on Machine Learning*, 1992.

[8] R. Kohavi and D. Sommerfield. Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. In *Proc. of the First Int'l Conference on Knowledge Discovery and Data Mining*, pages 192–197, Montreal, Quebec, 1995.

[9] I. Kononenko. Estimating attributes: Analysis and extensions of relief. In *Proc. of the 1994 European Conference on Machine Learning*, 1994.

[10] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, 1998.

[11] J. Moore, E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, and B. Mobasher. Web page categorization and feature selection using association rule and principal component clustering. In *7th Workshop on Information Technologies and Systems*, Dec. 1997.

[12] Hinrich Schtze, David A. Hull, and Jan O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Proceedings of SIGIR*, pages 229–237, 1995.

[13] P. Srinivasan. Retrieval feedback in medline. *Journal of the American Medical Informatiotrics Association*, 1996.

[14] Padmini Srinivasan. Query expansion and medline. *Information Processing & Management*, 32(4):431–443, 1996.

[15] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.

[16] Marilyn Wulfekuhler and Bill Punch. Finding salient features for personal web page categories. In *6th WWW Conference*, Santa Clara, CA, 1997.

[17] Y. Yang and J. Pederson. A comparative study on feature selection in text categorization. In *Proc. of the Fourteenth International Conference on Machine Learning*, 1997.